- TP n° 3 : Esterel -
MASTER SEMS
Conservatoire National des Arts et Métiers

lundi 12 novembre 2012

Objectif : L'objectif de ce TP est d'appréhender le langage Esterel en utilisant la version académique du compilateur v5_92. Cette version constitue l'implantation de référence de la sémantique v5 d'Esterel. À travers la réalisation de petits exemples de programmes on se familiarisera avec le langage et ses principaux outils : compilation, simulation, vérification.

Installation :

La première manipulation consiste à mettre en place l'environnement Esterel v5_92 : Téléchargez la distribution à l'adresse

```
http://www-sop.inria.fr/esterel-org/
```

Décompactez l'archive dans votre compte utilisateur. Il faut modifier le `Makefile` de sorte à ce que la variable `ESTEREL_DISTRIB_DIR` contienne le chemin d'accès effectif vers l'endroit où vous avez placé votre distribution Esterel. Lancez l'exécution du script `setup` puis lancez l'exécution de la commande `make` L'installation se termine par le positionnement de la variable d'environnement `PATH` : Ajouter dans votre `PATH` le chemin vers *<position du dossier Esterelv5_92>*`/bin`.
Vérifiez l'installation : entrez la commande

```
esterel -version
```

Le sous dossier `doc` contient un ensemble de fichiers au format PostScript ou PDF décrivant les principaux outils de la distribution (le compilateur Esterel, le simulateur XES et le vérificateur Xeve). Lire le manuel d'utilisation d'Esterel, pour prendre en main le compilateur et quelques unes de ses principales options. On s'intéressera particulièrement à la mise en œuvre des simulateurs (XES et en ligne de commande à l'aide de la bibliothèque csimul).

Quelques exemples simples :

Un exemple simple

Écrire un module qui attend un signal `Hello` et émet un signal `World`.

La compilation d'un tel module par défaut en esterel donne une fichier C que l'on doit intégrer dans une application qui fournira un contexte d'exécution (comment les signaux sont reliés à l'environnement

d'exécution). Écrire un fichier C contenant une fonction main et activant le module cycliquement. À la $10^{ème}$ activation, l'événement `Hello` est généré.

Commentez le comportement. Que se passe t'il si l'événement est généré au premier instant ?

Pour faciliter le test d'un module Esterel propose de créer un simulateur qui propose une interface pour tester différentes configuration événementielle. Mettez en oeuvre le simulateur `xes` puis le simulateur `csimul`.

ABRO

Reprendre l'exercice avec le module ABRO vu en cours, et tester les différences de comportement entre l'utilisation d'un `abort` et d'un `weak abort`.

Compilation en automate explicite

l'option `'-A'` du compilateur esterel permet de sélectionner la chaîne de compilation en automate explicite en produisant un fichier de type `.oc`. Recompilez les exemples de la séance précédente sous la forme d'automates explicites. Observez les différences de code. Pour stopper la génération automatique de code sur la production d'un code `.oc`, il suffit d'utiliser la directive `-oc`. Le fichier produit sera au format `.oc`. Observez un tel fichier.

Analyse de causalité

On souhaite compiler l'exemple suivant :

```
signal S1,S2 in
    emit S1;
    present S2 then
        present S1 else emit S1 end
    end
end
```

En utilisant la compilation sous forme d'un circuit séquentiel ce programme. Observer le comportement du compilateur.

Que se passe t'il pour la compilation en automate ? La compilation en automatique explicite fonctionne t'elle selon la sémantique v3 ?

On souhaite maintenant compiler le programme suivant :

```
signal S in
    present S then
        present S else emit S end
    end
end
```

Observez ce que produit le compilateur. Que se passe t'il si on utilise l'option -A.

Utilisation de l'outil Autograph

Récupérez l'outil autograph si ce n'est pas déjà fait, à l'adresse

`ftp://ftp-sop.inria.fr/meije/verif/atg/atgv3.linux.tar.gz`

Cet outil permet de explorer graphiquement des automates. Visualisez l'automate du programme ABRO. Pour cela il faut utiliser un format de représentation particulier : le format `.blif`. puis convertir à nouveau ce fichier dans le format `.fc2` utilisé par atg.

L'outil Xeve

le format `.bliff` est aussi utilisé par l'outil de vérification xeve. Mettre en œuvre cet outil sur l'exemple ABRO. Proposer une ou plusieurs propriétés à vérifier et implanter les observateurs synchrones correspondants.

Présentation d'un dispositif de suivi de trajectoire

Un système embarqué contient un module de positionnement physique de type GPS permettant de délivrer une information de position (latitude, longitude, altitude). Ces informations sont mises à disposition d'un module d'enregistrement qui sera en charge de les mémoriser. Comme aucune information de datation précise de ces informations de position n'est retournée par le module de positionnement, on utilise un troisième module qui fournit une horloge régulière pour cadencer le travail du module d'enregistrement. Ceci permettra de reconstruire la cinématique du système, basé sur une horloge en adéquation avec le mouvement de ce dernier (on se place dans le cadre de la mécanique classique).

Le module de positionnement

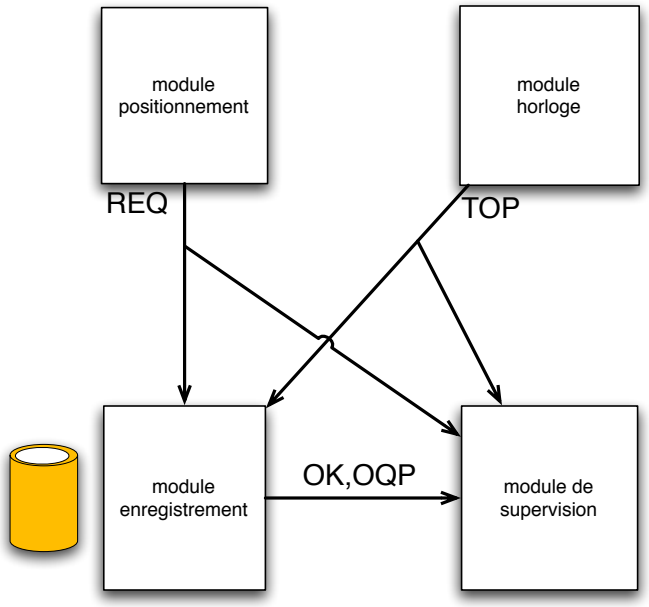
Il s'agit d'un dispositif qui calcul des informations de position sur le principe du GPS. Les informations ne sont pas émises de façon régulière car les calculs de positions n'ont pas forcément la même durée suivant la disponibilité des signaux de triangulation, etc. Aucune sortie permettant de dater précisément le calcul de position n'est fourni par ce module.

Le module d'enregistrement

Il s'agit d'un module qui se charge d'enregistrer l'historique des positions du système. Il prend en entrée les données du module de positionnement et les enregistre. Pour pouvoir reconstituer la cinématique du système, on lui adjoint un module horloge qui lui fournira un mécanisme de datation des positions. De sorte que périodiquement entre 2 tops d'horloge, le module n'enregistre qu'une et une seule information de position. La production des informations de position n'étant pas régulière, si plusieurs informations successives sont mises à disposition entre 2 tops d'horloge, seule la première sera enregistrée. Les informations suivantes seront omises et le module d'enregistrement indiquera au système cet état de fait.

module de supervision

Afin de vérifier que le processus de suivi de trajectoire est correct, un module de supervision sera en charge d'observer le système et de signaler un problème lorsque entre deux tops successifs du module d'horloge, aucune information de position n'a été proposée.



TOP/REQ :

module d'enregistrement

Il s'agit d'écrire un programme qui accepte une entrée `TOP` et une entrée `REQ`. Lorsque le test sur `REQ` est positif de nouvelles informations de position sont disponibles. Le test sur l'entrée `TOP` est positive à chaque top d'horloge. Entre 2 tops consécutifs le programme répond en positionnant `OK` à vrai ,à la première occurrence de `REQ`. Si un nouveau `REQ` survient entre 2 tops, le programme répond `OQP`.

module superviseur

Si entre deux tops successifs aucune requête n'a été faite au module d'enregistrement le module superviseur, lève une alarme : sortie `ALARM`

cas limite : Si `TOP` et `REQ` sont vrais simultanément alors la requête compte pour l'intervalle de temps (l'intervalle entre 2 `TOP`) qui débute et non pour celui qui termine.

Modélisation

Modéliser le système en Esterel. Simuler les programmes en utilisant les outils idoines aussi bien en mode textuel et graphique et décrire ce que vous obtenez. Votre programme fonctionne t'il correctement ? La compilation en automate de votre programme donne combien d'états ? Donnez une représentation graphique de l'automate du système résultant en utilisant l'outil `atg`.

Vérification

On souhaite vérifier que toute soumission d'une information de position est bien prise en compte (mais pas forcément enregistrée) par le module d'enregistrement. Comment procédez vous ? De même on souhaite vérifier qu'une alarme ne peut être émise que lorsqu'un top d'horloge est présent. Décrire ce que vous faites

et les résultats obtenus. Proposer également d'autres propriétés que vous pourriez vouloir vérifier sur votre système. Et commenter les résultats.

Module d'horloge

module d'horloge

Ce module sera construit à partir de l'horloge temps réel du système en utilisant les fonctions C du type `usleep` ou `gettimeofday` (La précision de la prise en compte du temps réel n'est pas la préoccupation principale ; il s'agit de simulation). On fixe l'intervalle au dixième de seconde. On souhaite également que le module d'horloge fournisse une information de datation. Il faut donc maintenant maintenir une information sur le nombre de seconde écoulées depuis la mise en route du système. Proposer une solution. Expliquer votre implantation.